METHOD AND APPARATUS FOR PROTECTING FILE SYSTEM BASED ON DIGITAL SIGNATURE CERTIFICATE

Field of the Invention

The present invention relates to a method and apparatus for protecting a file system; and, more particularly to a method and apparatus for protecting a file system based on a digital signature certificate in a computer system.

Prior Art of the Invention

In a conventional computer system, in order to protect a sever computer, an access control method or one-time password is used.

The computer system using the access control method permits a certain user to access only predetermined services or network addresses. In other words, the computer system prevents a user having no access authority from accessing except for the predetermined service or the predetermined network address.

A general password used for identifying a user is registered once and continually used until another password is registered. In order to prevent a malicious user from making a fraudulent use of the password, the one-time password is used. The one-time password means a password that is used only one time.

However, since a hacking method which a malicious hacker can obtain an authority of a system security manager or a general user by only accessing the predetermined service or network address has been introduced, interception of access to a certain service or network can not substantially protect the file system from the malicious hacker trying to forge or to alter the file system, e.g., a home page.

Various hacking techniques make partial security function of the one-time password powerless.

The problems of the access control method and one-time password result from a computer operating system that provides the conventional security technique implemented in application program, user or network level.

Summary of the Invention

Therefore, it is an object of the invention is to provide a method and apparatus for protecting a file system.

It is another object of the invention is to provide a safe and stable computer system.

In accordance with an aspect of the present invention, there is provided a method for protecting a file system in a computer, wherein a user having an access authority for a file can access the file system in the computer, the method comprising the steps of: a) generating system security manager's digital signature keys and system security manager's certificate; b) storing system security manager's certificate onto a security kernel when installing an operating system on a server computer; c) generating second digital signature keys and user's certificate; d) setting an access authority of the file system; e) identifying a user through a digital signature based authentication when the user tries to access the file system; and f) giving the user the access authority for the file in accordance with identification result.

In accordance with another aspect of the present invention, there is provided an apparatus for protecting a file system in a computer system, wherein a user having a file access authority can access the file system in the computer system, the apparatus comprising: means for generating system security manager's digital signature keys and system security manager's certificate; means for storing system security manager's certificate onto a

security kernel when installing an operating system on a server computer; means for generating user's digital signature keys and user's certificate; means for setting an access authority of the file system; means for identifying a user through a digital signature authentication method when the user tries to access the file system; and means for giving the user the access authority for the file in accordance with identification result.

Brief Description of the Drawings

The above and other objects and features of the instant invention will become apparent from the following description of preferred embodiments taken in conjunction with the accompanying drawings, in which:

Fig. 1 is a diagram of a computer system to which the present invention is applied;

Fig. 2 is a detailed diagram of a server computer in accordance with the present invention;

Fig. 3 is a detailed diagram of the security kernel of Fig. 2;

Fig. 4 is a detailed diagram of the certificate storage of Fig. 2;

Fig. 5 is a detailed diagram of the process security information storage in the security kernel of Fig. 3;

Fig. 6 is a detailed diagram of a file security information storage in the security kernel of Fig.3;

Fig. 7 is a flow chart illustrating a method for operating a file protecting method in accordance with the present invention;

Fig. 8 is a flow chart illustrating a method for installing the file protecting method on a server computer in accordance with the present invention;

Fig. 9 is a flow chart illustrating a method for operating the file protecting method in accordance with the present invention;

Fig. 10 is a flow chart illustrating a digital signature based authentication in accordance with the present invention;

Fig. 11 is a flow chart illustrating a method for registering/deleting a user in accordance with the present invention;

Fig. 12 is a flow chart illustrating a method for setting an access authority of a file in accordance with the present invention; and

Fig. 13 is a flow chart illustrating a method for processing a file.

Preferred Embodiments of the Invention

Hereinafter, preferred embodiments of the present invention will be described in detail with reference to the accompanying drawings.

Fig. 1 is a diagram of a computer system to which the present invention is applied.

The computer system includes a sever computer 110 and computers 120, 140 and 150 for a system security manager, a user at a remote distance and a user at a short distance from the server computer 110.

Each of computers 120, 140 and 150 has a storage device such like a floppy diskette 124, 144 and 154 and a smart card 126, 146 and 156. The sever computer 110 and the computers 120, 140 and 150 are connected to each other in direct or through a computer network 130.

The system security manager manages the sever computer 110 and users of the sever computers 110 after obtaining authentication based on digital signature.

The user 150 at a short distance from the sever computer 110 can access a part of files after being identified based on digital signature. The part of files are allowed to be accessed by the user. The system security manager sets an access authority of a file and an access

authority of a user.

The user 140 at a remote distance from the sever computer 110 can access a part of files allowed to be accessed by the user after being identified based on digital signature generated by communications through a computer network.

Fig. 2 is a detailed diagram of a server computer in accordance with the present invention.

The server computer includes a plurality of elements in a user level, a kernel level and hardware level.

The user level of the server computer includes a certificate storage block 212, a security management module 214, a security library 216, and a library 218.

The security management module 214 generates a pair of encryption key used for generating a digital signature value of the system security manager or the user at a short/remote distance. The pair of encryption keys includes a secret key and a public key. Also, the security management module 214 issues a certificate based on the encryption keys and the digital signature value.

The kernel level of the server computer includes a system call interface block 232, a file subsystem 234, a process control subsystem 236, a security kernel 238, a device driver 240 and a hardware controller 242.

The system call interface block 232 interfaces the elements in the user level with the elements of the kernel level.

The security kernel 238 verifies the digital signature, sets and inquires the access authority of the file. Also, the security kernel 238 controls an access of the file.

The hardware level of the server computer includes a driver controller, a hard disk driver, a floppy disk driver, a smart card driver, a Universal Serial Bus (USB) driver and a network driver.

These elements in the hardware level are well known to those skilled in the art. Accordingly, detailed description

on these elements will be skipped in this specification.

Fig. 3 is a detailed diagram of the security kernel of Fig. 2.

The security kernel includes an access authority control block 302, a digital signature verification block 304, an access authority setting/inquiry block 306, a security rule setting/inquiry block 308, a file system access authority decision block 310, a system security manager certificate storage 312, a process security information storage 314, a security rule storage 316 and a file system security information storage 318.

Security information related to the process is stored on the process security information storage 314, security rule information is stored on a security rule storage 316 and security information related to the file system is stored on a file system security information storage 318.

The access authority control block 302 controls the access authority setting/inquiry block 306, the security rule setting/inquiry block 308 and the file system access authority decision block 310.

The access authority setting/inquiry block 306 includes a process security information storage 320 and a file system security information setting/inquiry block 322. If a user trying to access a file is identified in the access authority control block 302, information on the process security information storage 314 is set by the process security information setting/inquiry block 320.

The file system security information setting/inquiry block 322 sets and inquires the file system security information storage 318.

The security rule setting/inquiry block 308 sets and inquires the security rules stored on the security rule storage 316.

The security rule setting/inquiry block 308 communicates with the access authority setting/inquiry block 306 and the file system access authority decision

block 310 and provides information necessary for an access control based on the security rules stored on the security rule storage 316.

The file system access authority decision block 310 compares information stored on the process security information storage 314 with the file system security information stored on the file system security information storage 318. The file system access authority decision block 310 determines whether an access authority is provided to the user based on the security rule stored on the security rule storage.

Fig. 4 is a detailed diagram of the certificate storage of Fig. 2.

The certificate storage 212 includes a plurality of certificates. The certificates include a user identification (ID) 410, 430 and 450 and a user certificate 420, 440 and 460. Each of the user identifications (ID) 410 represents the user possessing each of the user certificates 420. The pair of certificate is added, deleted or searched in accordance with a control signal from the security management module 214 of Fig. 2.

The user certificate 420 includes a system security manager identification (SM ID) 421, a user identification 422, an access authority identification (ID) 423, an access valid date 424, a public key 425, an issue time 426, a certificate valid date 427 and a digital signature value 428.

The system security manager identification 421 represents a system security manager SM who issues the user certificate.

The user identification 422 represents a user possessing the user certificate 420.

The access authority identification (ID) 423 represents an access authority of the user.

The access valid date 424 represents a valid time. The user can access the file system by the valid time.

The public key 425 is used for verifying a digital signature of a user. The issue time 426 represents a time on which the user certificate is issued.

The digital-signature value 428 represents a value digital-signed of the user certificate except the digital signature value 428 by using a secret key of the system security manager.

Fig. 5 is a detailed diagram of the process security information storage in the security kernel in Fig. 3.

In the process security information storage 314, a plurality of process identifications (ID) 510, system security manager flags 512 and access authority identifications (ID) 514 are stored. The process security information storage 314 searches the process identification 510 to be accessed. After finding the process ID, the process security information storage 314 sets or inquires a corresponding system security manager flag or access authority identification in accordance with a control signal from the process security information setting/inquiry block 320, the file system security information setting/inquiry block 322 or the file system access authority decision block 310.

Each of the process IDs 510 represents a process executed by the user.

Each of the system security manager flags 512 represents a system security manager by which a process is executed. Each of the access authority ID 514 represents an access authority permitted to the process.

Fig. 6 is a detailed diagram of a file system security information storage in the security kernel. The file system security information storage 318 of Fig. 3 includes a file identification (ID) 602 and an access authority identification (ID) 604. The access authority identification (ID) 604 corresponding to the file identification (ID) 602 is set or inquired in accordance with a control signal the file system security information

setting/inquiry block 322 or the file system access authority decision block 310.

The file identification (ID) 602 represents an identification used for identifying a file. The access authority identification (ID) 604 represents an access authority of the user allowed to access the file.

Fig. 7 is a flow chart illustrating a method for operating a file protecting method in accordance with the present invention.

First, an install process for setting a system security manager is performed at step 702. Next, an operating process is performed at step 704. In the operating process, a user registering/deleting process, a file access authority setting process or a file accessing process is performed after user authentication. Then, it is determined whether the file protecting method is terminated or not at step 706. If the method is not terminated, the process continues to step 704. If so, the method ends.

Fig. 8 is a flow chart illustrating a method for installing the file protecting method on a computer in accordance with the present invention.

First, the server computer generates a pair of keys for the system security manager, a public key PK_SM and a secret key SK_SM at step 802.

The sever computer generates a certificate for the system security manager at step 804. System security manager's access authority ACID_SM and system security manager's public key PK_SM are digital-signed by system security manager's secret key SK_SM, thereby generating the certificate for the system security manager.

The system security manager encrypts his/her secret key SK_SM and stores the encrypted secret key onto a memory device, e.g., a smart card or a floppy disk at step 806.

The system security manager stores his/her certificate CERT_SM onto a memory device, e.g., a smart card or a floppy disk at step 808. Also, the system security manager

stores his/her certificate CERT_SM onto the system security manager certificate storage 312 in the security kernel 238 at step 810.

The install process is terminated and returns to step 704.

Fig. 9 is a flow chart illustrating a method for operating the file protecting method in accordance with the present invention.

First, the server computer verifies a system security manager or a user trying to access itself by using digital signature based authentication at step 902. It is determined whether authentication result is success or fail at step 904.

If the authentication result is fail, the process terminates.

If the authentication result is success, the process goes to step 906 to load system security manager's certificate stored onto the system security manager certificate storage 312 and extracts system security manager's access authority ACID_SM from system security manager's certificate. And then, the process continues to step 908 to determine whether user's access authority ACID_U is equal to system security manager's access authority ACID_SM.

If user's access authority ACID_U is equal to system security manager's access authority ACID_SM, system security manager's access authority is applied to an access authority of a user process ACID_UP at step 910. The user process having system security manager's access authority ACID_SM selects and executes one of a user registering/deleting process, a file system access authority setting process and a file access process at steps 914, 916, 918 and 920.

If not, user's access authority ACID_U is applied to an access authority ACID_UP of a user process at step 912. The user process executes a file access process at step 920.

Then, the process returns to step 706.

Fig. 10 is a flow chart illustrating a digital signature based authentication process in accordance with the present invention.

The server computer generates a random number R at step 1002. The user generates a digital-signature value X to the random number R by using its secret key at step 1004. The server computer loads system security manager's certificate CERT_SM stored in the system security manager certificate storage in the security kernel 238 at step 1006. The server computer extracts the public key PK_SM of the system security manager from the certificate CERT_SM of the system security manager, the certificate CERT_SM being stored on the security kernel, at step 1008.

The certificate CERT_U of the user is verified by the security kernel 238 at step 1010. Then, it is determined whether verification result is success or fail at step 1012. If the verification result is fail, the process stores the verification result as a fail and terminates.

If the verification result is success, the security kernel extracts the public key PK_U and the access authority ACID_U of the user from the certificate CERT_U of the user at step 1014. After extracting the public key and the access authority for the client user, the security kernel verifies the digital signature value X to the random number R at step 1016. If an authentication result is success, the process stores the authentication result as a success and returns the access authority ACID_U of the user to the step 904 in order to be used at step 908.

Fig. 11 is a flow chart illustrating the user registering/deleting process in accordance with the present invention.

First, it is determined whether the access authority ACID_UP of the user process is equal to that ACID_SM of the system security manager at step 1102. If the access authority ACID_UP of the user process is not equal to that

ACID_SM of the system security manager, the process terminates and returns.

If the access authority ACID_UP of the user process is equal to that ACID_SM of the system security manager, the process continues the step 1104 to select a user registering process or a user deleting process.

If the user deleting process is selected, the user process having the access authority of the system security manager deletes the registered user at step 1106.

If the user registering process is selected, the user process having the access authority of the system security manager gives an access authority to a new user at step 1110. The user process generates a public key PK_U and a secret key SK_U for the new user at step 1112.

The system security manager encrypts the access authority and the public key for the new user by using its secret key, thereby generating a certificate CERT_U for the new user at step 1114. The new user encrypts its secret key and stores the encrypted secret key onto a memory device, e.g., a smart card or a floppy diskette at step 1116. The new user stores its certificate CERT_U onto the memory at step 1118. Then, it is determined whether the process is terminated or not at step 1120. If the process is terminated, the process returns. If not, the process goes to the step 1104 to select a user registering process or a user deleting process.

Fig. 12 is a flow chart illustrating a file access authority setting process in accordance with the present invention.

First, it is determined whether the access authority ACID_UP of the user process is equal to that ACID_SM of the system security manager at step 1202. If the access authority ACID_UP of the user process is equal to that ACID_SM of the system security manager, the system security manager selects a file of which access authority is to be set at step 1204. If not, the process terminates.

The system security manager selects users who are allowed to access the file at step 1204. The security kernel sets an access authority ACID_F of the file selected at the step 1204 as the access authority ACID_U of the user selected at the step 1206 at step 1208. Then, it is determined whether the process is terminated at step 1210. If the server computer selects termination, the process terminates. If not, the process continues to the step 1204.

Fig. 13 is a flow chart illustrating a method for processing a file.

The security kernel obtains a file to be accessed at step 1302. The security kernel compares the access authority ACID_UP of the user process trying to access the file with the access authority ACID_SM of the system security manager at step 1304.

If the access authority ACID_UP of the user process is equal to that ACID_SM of the system security manager, the server computer permits the user process to access the file F at step 1306. Then, it is determined whether the process is terminated. If the sever computer selects termination, the process terminates. If not, the process continues to the step 1302.

If the access authority of the client user process is not equal to that of the system security manager, the process goes to step 1308 to determine whether the access authority of the user process ACID_UP is equal to that ACID_UP of the user at step 1308. If not, the process terminates.

If the access authority ACID_UP of the user process is equal to that ACID_SM of the user, it is determined whether the access authority ACID_UP of the user process is equal to that ACID_F of the file F at step 1310. If not, the process terminates.

If the access authority ACID_UP of the user process is equal to that ACID_F of the file F, the server computer permits the user process to access the file at step 1312.

Then, it is determined whether the process is terminated. If the server computer selects termination, the process terminates. If not, the process continues to the step 1302.

The file protecting system in accordance with the present invention stores the certificate of the system security manager onto the security kernel in the kernel level at system install process. Also, the digital signature based authentication, a file access authority setting process and a file access process are performed in the kernel level not in the user level. Accordingly, the file protecting system can fundamentally prevents the file system from being forged or altered.

Therefore, the file protecting system in accordance with the present invention provides a stable and reliable file system. For example, the file protecting system in accordance with the present invention can protect a web server system operating a homepage from a hacking.

Although the preferred embodiments of the invention have been disclosed for illustrative purpose, those skilled in the art will be appreciate that various modifications, additions and substitutions are possible, without departing from the scope and spirit of the invention as disclosed in the accompanying claims.